UIUCDCS-R-79-981 UILU-ENG 79 1727

Canonical Simplification of Finite Objects Well Quasi-Ordered by Tree Embedding

by

Thomas C. Brown, Jr.

August 1979

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

Canonical Simplification of Finite Objects[†]
Well Quasi-Ordered by Tree Embedding

THOMAS C. BROWN, JR.[††]

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois

March 1979

[†] This work was supported in part by the National Science Foundation under grant NSF MCS 77-22830.

[††] Author's mailing address: Computer Science Department, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

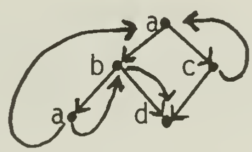
ABSTRACT

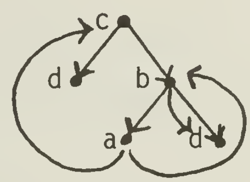
A finite object space can be viewed as the set of finite and infinite (edge-) ordered trees representable as finite ordered vertex-labeled digraphs. We show that the order-preserving homeomorphic tree-embedding relation on finite objects over a well quasi-ordered (wqo) label alphabet is wqo. Canonical simplifiers require well-founded object-orderings to orient and ensure finite termination of replacement rules. We characterize these orderings as homomorphic refinements of the wqo tree-embedding relation, and we show this relation's time complexity to be O(mxn) where m and n are the edge counts of the digraph representations.

0. Introduction

Conceptually infinite objects arise naturally in λ-calculus and related programming formalisms familiar to computer scientists [1]. Many of these objects - e.g., effective procedures and the objects they process - have finite representations as (edge-) ordered digraphs with vertex labels in some previously defined finite object space. Canonical simplifiers and other procedures use well-founded orderings on such spaces to orient their replacement rules and ensure termination of their computations. This note establishes a fundamental connection between these orderings and a homeomorphic embedding relation on the finite or infinite ordered tree representations of finite objects. A quadratic time-bounded embedding procedure and a λ-calculus application are included to illustrate the practical significance of this connection.

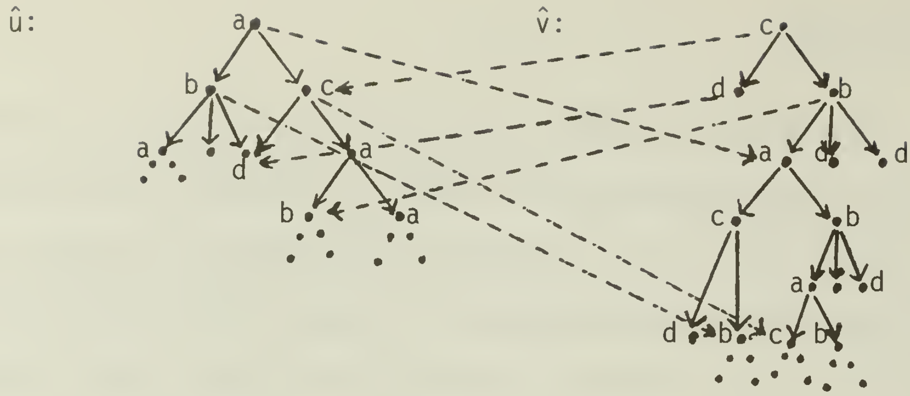
The concepts involved are illustrated by the following pair of labeled digraphs:

u:


v:


(1)

These objects may represent recursive program schemas with conditional branches,or terms in a λ-applicative syntax with fixed point operators [§3]. Incomparable under the (NP-complete [10]) graph-embedding relation, they are equivalent under the induced order of homeomorphic embedding on their infinite ordered tree representations [§1.13]:



The indicated vertex correspondence shows the initial part of an embedding $h\colon \hat{u} \tilde{<} \hat{v}$ as defined precisely in §1. It preserves the label ordering (identity in this case) on corresponding vertices, and it embeds distinct branches descending from a vertex in distinct branches descending from the corresponding vertex, in accordance with the orderings on their outgoing edge sets. Homeomorphic embedding and equivalence are decided in $O(\langle u \rangle \times \langle v \rangle)$ elementary operations by the procedure described in §2, where $\langle u \rangle$ is the number of edges in the finite digraph representation of u.

The basic result of §1 is that finite objects over a well quasi-ordered (wqo [18]) vertex label alphabet are wqo by the tree embedding relation. A quasi-ordered set $(Q, \tilde{<})$ (where $\tilde{<}$ is reflexive and transitive) is wqo if

(i) it is well founded: $q_k \tilde{>} q_{k+1} (k \varepsilon \omega)$ implies $q_k \tilde{<} q_{k+1} (k \geq n)$ for some number $n \varepsilon \omega$; and

(ii) it admits no infinite antichains $A \subseteq Q$, where $u \not\gtrless v (u, v \varepsilon A)$.

(iii) every infinite sequence $q\colon \omega \to Q$ has an infinite subsequence $(q_{j_k} : k \varepsilon w)(j_k < j_{k+1})$ where $q_{j_k} \tilde{<} q_{j_{k+1}} (k \varepsilon \omega)$.

Thus every infinite sequence of finite digraph-representable trees over a finite (or more generally, wqo) alphabet has an infinite homeomorphic embedding subsequence.

This result extends Kruskal's tree theorem [17] from finite trees to finitely representable trees by a straightforward extention of Nash-Williams' simplified finite-tree argument [26]. Thus it strengthens the general maxim that finite object spaces constructed reasonably from wqo building blocks are wqo. For transfinite objects (infinite sets, sequences, or trees) the maxim fails [32]. On the basis of Rado's and other counterexamples due to Kruskal, Nash-Williams identified a restricted class of wqosets which are preserved under many transfinite object space constructions [27, 28]. These better quasi-ordered sets (bqosets) include all finite quosets and all of the finite object spaces over bqosets considered below. Laver [21, 22, 23] has extended Nash-Williams' results to transfinite ω-level bounded (and even deeper) trees over a bqoset (without the edge-order preservation constraint), and has since refined these arguments to the class of embeddings which respect a bqo-labeled linear ordering (of a quite general type [21]) on the edges or branches leaving each vertex.[1]

The wqo preservation argument for finite object spaces in §1 is of independent interest for its relative simplicity, its corollary complexity bounds, and its applications to canonical simplification. Specifically, we show in §3 that an equationally generated reducibility relation on a class of λ-applicative terms is well founded iff the replacement rules refine a quasi-simplification ordering (qso) on these terms. Well foundedness of qso's on terms over a wqo (sub-)vocabulary follows by the main result of §1. These results hold for the initial "rationally closed" algebra [8] with cyclic terms. They complement previously investigated methods for reasoning about and processing finite (possibly cyclic) objects [9, 24, 25, 38], and they facilitate construction and recognition of canonical (rule based) simplifiers with the Church-Rosser property [2, 15, 16, 19].

1. Well Quasi-Ordering by Tree Embedding

By a space we normally mean a decidable (countably) infinite set of finite objects; such spaces are recursively isomorphic to the natural numbers [35]. Specifically, given a quoset $\underline{\Sigma} = (\Sigma, \tilde{<})$ where $\Sigma \subseteq$ some space, we are interested in the space R_Σ consisting of a representative (up to isomorphism) of each finite rooted edge-ordered digraph with vertex labels in Σ. Tree embedding is most naturally defined below on the set $\hat{R}_\Sigma$ of labeled trees representing R_Σ; the induced relation on R_Σ is easily computed [§2]. R_Σ (or a "well formed subspace" [§3]) is a suitable carrier for the initial rationally closed algebra [11,37] or rational object space [8] over Σ.

Conventions. Metavariables over R_Σ:r, s, t, u, v; over Σ:a, b, c, d. Occasionally we identify c in Σ with its edgeless singleton digraph ($\Sigma \subseteq R_\Sigma$). $\tilde{>}$ = converse of $\tilde{<}$; $x < y$ iff $x \tilde{<} y \tilde{\not>} x$; $x \simeq y$ iff $x \tilde{<} y \tilde{<} x$. $(Q, \tilde{<})$ is the quoset based on a specified extension of $\tilde{<}$ from Σ to Q. ω = finite ordinals: $n^+ = n \cup \{n\} = n+1$; $n^- = m$ where $m^+ = n$; $0^- = 0 = \{\ \}$. $|X|$ = cardinality of X. X^* = free nonoid over X (concatenation "$\cdot$"); $X \subseteq X^*: \langle x \rangle = x$, $\langle\ \rangle$ = null sequence. Metavariables over ω^* : $\alpha, \beta, \gamma, \delta$.

The following "data type algebra" provides a convenient formal basis for reasoning about R_Σ:

1.1 Definitions. $(R_\Sigma \vee \{\perp\}, \Sigma, \delta, \lambda, \omega)$ is a two-sorted algebra where

(i) $\delta(u)$ = out-degree of root vertex of u;

(ii) $\lambda(u)$ = label of root vertex of u (in Σ); and equivalently

(iii) $u_{\langle k \rangle} = \begin{cases} \text{k-th (immediate) constituent (in } R_\Sigma\text{) of u, if } k \in \delta(u); \\ \perp\text{, otherwise } (k \in \omega). \end{cases}$

As a technical convenience we also define

(iv) $\rho(u)$ = root vertex of u.

While not explicitly concerned with vertices, we find it convenient to assume them to be natural numbers in §2. We extend (iii) inductively to ω^*:

(v) $u_{\alpha.k} = (u_\alpha)_{\langle k \rangle}$, if $u_\alpha \in R_\Sigma$;

(vi) $D(u) = \{\alpha \in \omega^*: u_\alpha \in R_\Sigma\}$.

D(u) is a finitary tree domain ($u \in R_\Sigma$):

1.2 Definition. A tree domain is a set $D \subseteq \omega^*$ such that

(i) $\langle\ \rangle \in D$; and

(ii) $\alpha.k \in D$ implies $\alpha.k^-$, $\alpha \in D$.

It is finitary if $\{k: \alpha.k \in D\}$ is finite $(\alpha \in D)$. Given $u \in R_\Sigma$, $\hat{u}:D(u) \to \Sigma$ is the labeled tree defined by $\hat{u}(\alpha) = \lambda(u_\alpha)$. $\hat{R}_\Sigma = \{\hat{u}:u \in R_\Sigma\}$.

1.3 Definitions. A vertex $\rho(u_\alpha)$ in u is said to be cyclic if $u_{\alpha.\beta} = u_\alpha$ for some $\beta \neq \langle\ \rangle$; otherwise acyclic. We may speak of $\alpha.\beta$ as a cycle in u with root α. The edge represented by a path $\alpha.k$ in u is the k-th directed arc, which connects u_α and $u_{\alpha.k}$. It is said to cyclic if $u_{\alpha.k.\beta} = u_\gamma$ for some β and some prefix γ of α; otherwise acyclic. (A cyclic edge or vertex may be indexed several times by the prefix set of a cycle $\alpha.\beta$ in u.)

1.4 <u>Definitions</u>. Distinct constituents u_α, u_β (or their roots) are said to be <u>mutually</u> <u>connected</u> if $u_{\alpha.\beta'} = u_\beta$ and $u_{\beta.\alpha'} = u_\alpha$ for some α', β', in which event $\alpha.\beta'.\alpha'$ and $\beta.\alpha'.\beta'$ are cycles with respective roots α, β. The (unique maximal strongly connected [30]) <u>component</u> of u consists of u ($\rho(u)$) and all constituents (vertices) mutually connected with u (with associated edges).

$SC(u) =_{df} \{\beta \in D(u)\colon u_\beta$ is mutually connected with $u\}$ thus $\{SC(u_\alpha)\colon \alpha \in D(u)\}$ partitions $D(u)$ into tree-address sets of distinct components.

1.5 <u>Corollary</u> [30]. Each object in R_Σ has a unique root component, from which every other component is accessible by a path through some acyclic edge leaving (a member of) that component.

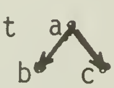
1.6 <u>Definition</u>. Let $\alpha \wedge \beta$ = longest common prefix of α and β in ω^*. A mapping $h\colon\ D(u) \to D(v)$ is said to be a (homeomorphic) <u>embedding</u> of $\hat{u}$ in $\hat{v}$ ($h\colon \hat{v} \tilde{\leq} v$) if

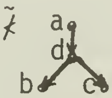
(i) $h(\alpha \wedge \beta) = h(\alpha) \wedge h(\beta) (\alpha.\beta \in D(u))$;

(ii) $\alpha.j \in D(u)$ and $0 \leq i < j$ imply existence of $i' < j'$, β, γ such that $h(\alpha.i) = h(\alpha).i'.\beta$ and $h(\alpha.j) = h(\alpha).j'.\gamma$; and

(iii) $\lambda(u_\alpha) \tilde{<} \lambda(u_{h(\alpha)}) (\alpha \in D(u))$.

We define $u \tilde{<} v$ iff $h\colon \hat{u} \tilde{<} \hat{v}$ for some embedding h.

Note that 

$\tilde{\not<}$ 

by this definition (unless $a \tilde{<} d$); the more general embeddings are admitted if (i) is replaced by "$h(\alpha \wedge \beta)$ is a prefix of $h(\alpha) \wedge h(\beta)$". (That R_Σ is wqo by these is an immediate consequence of Theorem 1.7.)

Now we can state the main result of this section more precisely:

1.7 <u>Theorem</u>. $(\Sigma, \tilde{<})$ wqo implies $(R_\Sigma, \tilde{<})$ wqo.

The following lemma summarizes some well-known properties of wqo sets. These follow from the definitions by sequence-refinement arguments. An infinite sequence $\underset{\sim}{q}$ over Q is said to be <u>good</u> if $q_i \tilde{<} q_j$ for some $i < j$ (otherwise <u>bad</u> [26]), and <u>ubiquitous</u> if is has an infinite $\tilde{<}$ - ordered subsequence $(q_{j_k} \tilde{<} q_{j_{k+1}}, j_k < j_{k+1}, k \in \omega)$.

1.8 <u>Lemma</u>.

(a) $\underline{Q}$ is wqo iff each infinite sequence over Q is good iff each infinite sequence over Q is ubiquitous.

(b) $(Q_0 \cup Q_1, \tilde{<})$ is wqo iff $\underline{Q_0}$ and $\underline{Q_1}$ are wqo.

(c) If $(Q_i, \tilde{<}_i)$ is wqo $(i = 0, 1)$ then $Q_0 \times Q_1$ is wqo with the product ordering: $(q_0, q_1) \tilde{<} (q_0', q_1')$ iff $q_i \tilde{<}_i q_i'$ $(i = 0, 1)$.

(d) If $h\colon \underline{Q} \to \underline{R}$ is a qoset homomorphism ($x \tilde{<} y$ implies $h(x) \tilde{<}_R h(y)$ and $h[Q] = R$ then $\underline{Q}$ wqo implies $\underline{R}$ wqo.

Now to prove Theorem 1.7 we use the first of several applications of Nash-Williams' "minimal bad sequence" argument [26]. Suppose $\underline{\Sigma}$ wqo and $\underline{R_\Sigma}$ not wqo. Let u_0 be an irreducible $\tilde{<}$ - minimal element which begins some bad sequence over R_Σ. Let u_{k+1} be an irreducible $\tilde{<}$ - minimal object which extends $(u_0, \ldots, u_k)$ to a prefix of some bad sequence over R_Σ. Let W be the set of (maximal) non-root component constituents of elements of the bad sequence $\underset{\sim}{u} = (u_k\colon k \in \omega)$.

1.9 Lemma. $(W, \tilde{<})$ is wqo.

Proof. If W is finite this is obvious. Otherwise refine some bad sequence over W to obtain a bad $\underset{\sim}{t} = (t_k: k \in \omega)$ such that

(i) t_0 is a (non-root component) constituent of u_j; and

(ii) $t_k \not\tilde{>} u_i$ $(i = 0, \ldots, u_j;\ k > 0)$.

Then $(u_0, \ldots, u_j, t_0, t_1, \ldots)$ is bad because $u_i \tilde{<} t_k$ implies $u_i \tilde{<} u_n$ for some u_n where $n \geq j$ (contra choice of u_n). However, $t_0 < u_j$ contradicts the choice of u_j in $\underset{\sim}{u}$. ■

Remarks. Irreducible objects were not essential here; they are required in subsequent "minimal bad sequence" arguments which split components. The lemma suggests that we view each element of $\underset{\sim}{u}$ as a root component whose outgoing acyclic edges (if any) terminate in a new wqo vocabulary $(W, \tilde{<})$. Formally, transform u_k into an object u_k' over $\Sigma \cup W$ with cyclic root-component labels in Σ and adjacent acyclic vertices labels in W. Now observe that $u_j' \tilde{<} u_k'$ implies $u_j \tilde{<} u_k$: indeed, there exists $h: \hat{u}_j \to \hat{u}_k$ where h embeds $SC(u_j)$ in $SC(u_k)$.

1.10 Definitions. The component height $CH(u)$ is defined inductively by

$$CH(u) = \operatorname{Sup}\{CH(u_\alpha): \alpha \in D(u) - SC(u)\}.$$

$$S_\Sigma = \{u \in R_\Sigma: CH(u) \leq 1 \text{ and all cyclic vertices of } u \text{ are in its root component}\}$$

S_Σ contains all strongly connected $u \in R_\Sigma$ $(CH(u) = 0)$, and $S_{W \cup \Sigma}$ contains all elements u_k' (Remark). Now observe that the alleged bad sequence is refuted (proving Theorem 1.6) by the following (with $\Sigma \cup W$ for Σ):

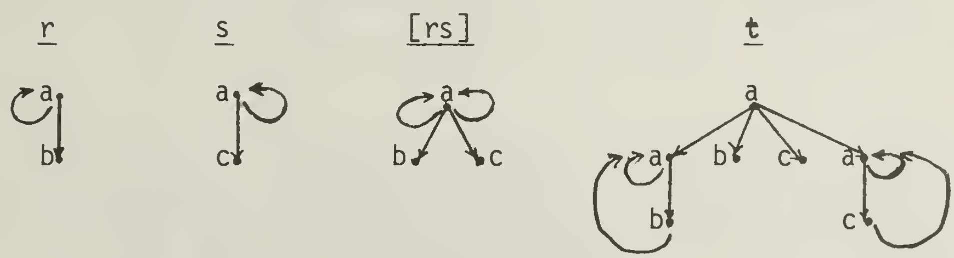
1.11 <u>Theorem</u>. $\underline{\Sigma}$ wqo implies $(S_\Sigma, \underline{<})$ wqo.

Indeed, from this result and the ubiquity property of wqosets [Lemma 1.8(a)] we infer the following strengthening of Theorem 1.8 by another "minimal bad sequence" argument:

1.12 <u>Corollary</u>. $\underline{\Sigma}$ wqo implies that for each infinite sequence $\underset{\sim}{u}$ over R_Σ there is an infinite subsequence $(u_{j_k} : k \in \omega)$ wherein $u_{j_k} \tilde{<} u_{j_{k+1}}$ by an embedding h: $\hat{u}_{j_k} \tilde{<} \hat{u}_{j_{k+1}}$ which maps $SC(u_{j_k})$ into $SC(u_{j_{k+1}})$ - i.e., h "embeds the root component of u_{j_k} in the root component of $u_{j_{k+1}}$."

<u>Proof</u>. Suppose not; let $\tilde{u}$ be a $\tilde{<}$ - minimal sequence for which the alleged embedding subsequence does not exist, and obtain $\underline{W}$ (wqo <u>by</u> Theorem 1.7) as in Lemma 1.9. By the preceding remark applied to an infinite embedding subsequence of $(u_k' : k \in \omega)$ over $S_\Sigma \cup W$ we obtain the desired subsequence of $\underset{\sim}{u}$.■

The remaining "minimal bad sequence" argument for Theorem 1.11 involves analysis and synthesis of components. It may be helpful to observe at the outset that embeddings of parts yield embeddings of wholes in S_Σ but <u>not</u> in R_Σ:



(2)

That $r \tilde{<} t$, $s \tilde{<} t$, and $[rs] \tilde{\not<} t$ has no bearing on arguments below because $t \notin S_\Sigma$ (even though CH(t) = 1). The invariance of $\tilde{<}$ under analysis and synthesis operations on S_Σ is clarified by the following representation for $(S_\Sigma, \tilde{<})$.

1.13 <u>Definitions</u>. A <u>cell</u> over Σ is a word $cw \in \Sigma \times (\Sigma \cup \{\tau\})^*$ where τ is a new symbol, $\tau \tilde{<} \tau$, and τ is unrelated to elements of Σ by $\tilde{<}$. cw is said to be <u>cyclic</u> if w contains ; otherwise <u>acyclic</u>. Let $\tilde{S}_\Sigma$ be the space of finite nonempty cell sets, each either a singleton or a set of cyclic cells. Define $\tilde{u}$ in $\tilde{S}_\Sigma$ for u in S_Σ by

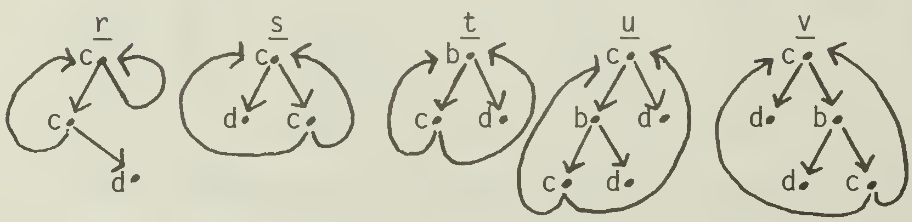
$$\tilde{u} = \begin{cases} \{\lambda(u)\}, \text{ if } \delta(u) = 0; \\ \{\langle \lambda(u)\ \lambda(u_0)\ \ldots\ \lambda(u_{\delta(u)^-})\rangle\}, \text{ if } u \text{ is acyclic, } \delta(u) > 0; \\ \{\langle \lambda(u_{\alpha_k})\ e_0^k \ldots e^k_{\delta(u_{\alpha_k})^-}\rangle : k \in n\}, \text{ otherwise,} \end{cases}$$

where u has root component constituents $\{u_{\alpha_0}, \ldots, u_{\alpha_{n^-}}\}$ and

$$e_j^k = \begin{cases} \lambda(u_{\alpha_k . j}),\ \delta(u_{\alpha_k . j}) = 0; \\ \tau, \text{ otherwise } (j \in \delta(u_{\alpha_k}) > 0). \end{cases}$$

Define $\tilde{u} \tilde{<} \tilde{v}$ iff to each cw in $\tilde{u}$ there corresponds $c'w'$ in $\tilde{v}$ such that $c \tilde{<} c'$ and if $|w| > 0$ then for some $\eta \in \omega^*$ where $0 \leq \eta_0 < \ldots < \eta_{|w|^-} \leq |w'|^-$ we have (for all $k \in |w|$) either $w_k \tilde{<} w'_{\eta_k}$ or $w_k \tilde{<} a$ where $w'_{\eta_k} = \tau$ and a occurs in some member of $\tilde{v}$.

<u>Examples</u>. For u, v in (1) we have $\tilde{u} = \{a\tau\tau, b\tau dd, cd\tau\} = \tilde{v}$. However, $\underline{\tilde{S}_\Sigma}$ is not a canonical representation for $(S_\Sigma, \tilde{<})$:

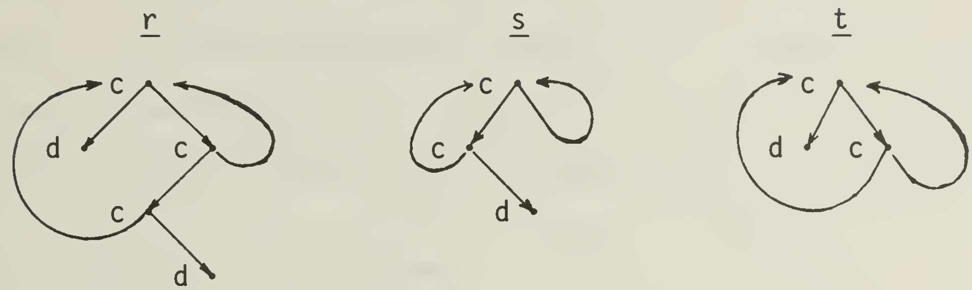


(3)

Here we have $\tilde{r} = \{c\tau\tau, c\tau d\} \simeq \tilde{s} = \{cd\tau, c\tau\tau\} > \tilde{t} = \{b\tau d, c\tau\tau\} \simeq \tilde{u} = \{c\tau d, b\tau d, c\tau\tau\} \simeq \tilde{v} = \{cd\tau, bd\tau, c\tau\tau\}$. For example $\tilde{s} < \tilde{t}$ because both $cd\tau$ and $c\tau\tau$ match $c\tau\tau$ in $\tilde{t}$ due to the presence of d in $b\tau d$.

1.14 Definition. $u(\tilde{u})$ is said to be reducible if $\{x\} \precsim \tilde{u} - \{x\}$ for some cell x in $\tilde{u}$; otherwise irreducible.

Thus $\tilde{u}$ and $\tilde{v}$ in (3) both reduce to $\tilde{t}$. Reduction preserves $\simeq$; however, an object can have distinct irreducible reducts:



Here we have $\tilde{r} = \{ cd\tau, c\tau\tau, c\tau d\} \simeq \tilde{s} = \{c\tau\tau, c\tau d\} \simeq \tilde{t} = \{c\tau\tau, cd\tau\}$.

1.15 Lemma. $(\tilde{S}_\Sigma, \precsim)$ represents $(S_\Sigma, \precsim)$, and embeddings of cyclic pairs imply embeddings of their joins in $\underline{S}_\Sigma$ $(\tilde{S}_\Sigma)$:

(a) $u \precsim v$ iff $\tilde{u} \precsim \tilde{v}$;

(b) if s and t are cyclic and $(s, t) \precsim (s', t')$ then

$$\tilde{s} \cup \tilde{t} \precsim \tilde{s}' \cup \tilde{t}'.$$

Consequently, $u \simeq u'$ for some irreducible u', and $t \subsetneq u'$ implies $t \subsetneq u$ $(t, u \in S_\Sigma)$.

Proof. If u or v is acyclic, then (a) is obvious. Suppose u, v cyclic with respective root component sets $\{u_{\alpha_k} : k \in m\}$, $\{v_{\beta_k} : k \in n\}$. Let $\{x_k : k \in m\}$ be the cells corresponding to $\{u_{\alpha_k} : k \in m\}$, and $\{y_k : k \in n\}$ the cells corresponding to $\{v_{\beta_k} : k \in n\}$.

Suppose $h : \hat{u} \precsim \hat{v}$. Then $v_{h(\alpha_i)} = v_{\beta_j}$ for some j, establishing a functional embedding $M(x_i, y_j)$ of $\tilde{u}$ into $\tilde{v}$ satisfying Definition 1.13.

Conversely, if $M : \tilde{u} \to \tilde{v}$ is such an embedding, then we can define $h : \hat{u} \to \hat{v}$ by induction; $u = u_{\alpha_i}$ for some i, whence $h\langle\rangle = \beta_j$ where $M(x_i, y_j)$. Given $h(\alpha)$ where $u_\alpha = \alpha_i$ we must have $v_{h(\alpha)} = v_{\beta_j}$ for some j by definition of M, whence the extension of h below α based on Definition 1.13. (Consider the three cases for $k \in \delta(u_{\alpha_i})$, $x_i = cw$, $y_j = c'w'$, η as stipulated: (i) $w_k \precsim w'_{\eta_k} \in \Sigma$; (ii) $w_k = \tau = w'_{\eta_k}$; and (iii) $w_k \precsim a$ where $w'_{\eta_k} = \tau$ and a occurs in some cell of $\tilde{v}$. Definition of $h(\alpha.k)$ is straightforward in each case - e.g., in (iii) let $h(\alpha.k) = h(\alpha).\eta_k.\gamma$ where $\lambda(v_{\beta_j.\eta_k.\gamma}) = a$.).

(b) Given $s \precsim s'$ and $t \precsim t'$ where s and t are cyclic, it follows by (a) that s' and t' are cyclic and $\tilde{s} \cup \tilde{t}$, $\tilde{s}' \cup \tilde{t}'$ are sets of cyclic cells. Moreover, matchings $M_s : \tilde{s} \to \tilde{t}'$ and $M_t : \tilde{t} \to \tilde{t}'$ define a matching $M_s \cup M_t$: $\tilde{s} \cup \tilde{t} \to \tilde{s}' \cup \tilde{t}'$.

Now suppose $\{x\} \precsim \tilde{u} - \{x\}$ where $x \in \tilde{u}$. Then u and x are both cyclic, whence $\tilde{u} = \{x\} \cup (\tilde{u} - \{x\}) \precsim \tilde{u} - \{x\}$ by (b). Thus $\tilde{u} \simeq \tilde{u} - \{x\}$, whence $\tilde{u} \simeq \tilde{u}'$ for some irreducible u'. If $\tilde{t} \cup \{x\} \subseteq \tilde{u}'$ where $x \notin \tilde{t}$ then $t \precsim u' \simeq u$, whence $t < u' \simeq u$ by irreducibility of u'.∎

We conclude the proof of Theorem 1.11 by supposing Σ wqo (wpo) and u an irreducible $\precsim$ - minimal bad sequence over S_Σ. Let $\{u_k : k \in w\} = \mathcal{U} \cup \mathcal{V} \cup \mathcal{W}$ where

$$\mathcal{U} = \{u_k : \delta(u_k) = 0\};$$
$$\mathcal{V} = \{u_k : \delta(u_k) > 0 \wedge |\tilde{u}_k| = 1\};$$
$$\mathcal{W} = \{u_k : |\tilde{u}_k| > 1\}.$$

By Lemma 1.8 (a,b) it suffices to show that $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ are wqo. For $\mathcal{U} \subseteq \Sigma$ this is immediate.

Let $V' = \{u \in V : \delta(u) = 1\}$, $V'' = \{u \in V : \delta(u) > 1\}$. V' consists of "monadic loops" $c\circlearrowleft$ and "pairs" $\begin{smallmatrix} c \\ \downarrow \\ d \end{smallmatrix}$; both subsets are evidently wqo (by Lemmas 1.15 and 1.8(c), respectively). From V'' we obtain $S = \{s_k : k \in w\}$ by retaining only first edges of roots of elements of V'' (wqo as for V'), and T by deleting first edges of roots of elements of V''. Then $t_k < u_k$ $(u_k \in V'')$, and T is wqo by the same argument used for Lemma 1.9. Thus $S \times T$ is wqo. Suppose this set is infinite. Then $(s_i, t_i) \precsim (s_j, t_j)$ for some $i < j$ [Lemma 1.8(a)]. This implies $u_i \precsim u_j$, a contradiction. ($u_i = [s_i t_i]$ and $u_j = [s_j t_j]$ as in (2); consider the two cases : s_j cyclic, s_j acyclic.) Therefore, V'' is finite and V is wqo.

From W we obtain $S = \{s_k : u_k \in W\}$ and $T = \{t_k : u_k \in W\}$ where $s_k, t_k < u_k$ and $\tilde{u}_k = \tilde{s}_k \cup \tilde{t}_k$. Again by definition of $\underset{\sim}{u}$, we have S, T wqo whence W is finite. (Otherwise, there exist $i < j$ where $(s_i, t_i) \precsim (s_j, t_j)$; apply Lemma 1.15.)

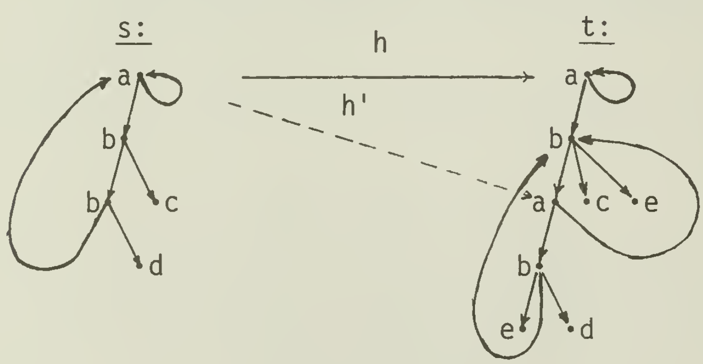
This concludes the proof of Theorem 1.11. The preceding case analysis generalizes earlier arguments used to show that Σ^* and the space of finite subsets of Σ are wqo by natural extensions of $\precsim$ [17].

2. Quadratic Time Bounds

This section describes an algorithm for $(R_\Sigma, \precsim)$ which decides $(u \precsim v)$ in $O(\langle u\rangle \times \langle v\rangle)$ elementary assignment, indexing, and scalar comparison operations, where $\langle u\rangle$ = number of edges in u. The algorithm operates by associating with each (strongly connected) component of v a record of all components of u which can be embedded within or below that component. It is similar to bottom-to-top occurrence-finding procedures [14]. Its correctness proof is based in part on the following "strong embedding" refinement of $\precsim$ [§1.6, §1.10].

2.1 Definition. An embedding $h : \hat{u} \to \hat{v}$ is strong if $h[SC(u_\alpha)] \subseteq SC(v_{h(\alpha)})$ $(\alpha \in D(u))$.

Remark. In other words, h "embeds components within components." Not all embeddings are strong:



(4)

Evidently, h does not map $SC(s) = \{\langle\ \rangle, 0, 0.0, 0.0.0, 1,\ldots\}$ into $SC(t) = \{\langle\ \rangle, 1,\ldots\}$. However, such an embedding can always be constructed from a given embedding.

2.2 Lemma. If $u \lesssim v$, then there exists a strong embedding h': $u \lesssim v$.

Proof. Consider u and v as trees of strongly connected components (with cyclic edges), joined by acyclic edges. The argument is by induction on CH(u).

Suppose $h[SC(u)] \not\subseteq SC(v_{h<>})$. Then SC(u) is infinite; there must be some $\beta' \in$ Dom (v) such that $h[SC(u)] \cap SC(v_{\beta'})$ is infinite. Choose $\alpha \in SC(u)$ such that $h(\alpha) \in SC(v_{\beta'})$. There is β in SC(u) such that $u_{\alpha.\beta} = u$. It follows that $h(\alpha.\beta.\gamma) \in SC(v_{\beta'})$ for all $\gamma \in SC(u)$. (h[SC(u)] cannot have infinite intersections with two distinct SC-classes in Dom (v).) Thus, let $h'< > = h(\alpha.\beta)$, and let $h'(\gamma) = h(\alpha.\beta.\gamma)$ ($\gamma \in SC(u)$).

Now consider a path $\delta.k$ in u where $\delta \in SC(u)$ and $\rho(u_{\delta.k})$ is not connected to $\rho(u)$. Then, h embeds $u_{\delta.k} = u_{\alpha.\beta.\delta.k}$ in $v_{h(\alpha.\beta.\delta.k)}$ (where $h(\alpha.\beta.\delta.k)$ may or may not be in $SC(v_{\beta'})$), and $CH(u_{\delta.k}) < CH(u)$. It follows by the induction hypothesis that we can extend h' to embed $u_{\delta.k}$ strongly in $v_{h(\alpha.\beta.\delta.k)}$ for each such path $\delta.k$.∎

Given the sufficiency of a strong embedding test and the apparent effectiveness of $(S_\Sigma, \lesssim)$ as a representation for $(S_\Sigma, \lesssim)$, it is only natural to consider an algorithm for $(u \lesssim v)$ in $\underline{R}_\Sigma$ which reduces this problem to simpler decisions $(\tilde{s} \lesssim \tilde{t})$ in $\tilde{S}_{\Sigma'}$. The algorithm for $\underline{R}_\Sigma$ is based on the following "component representation," essentially an extension of ~ from S_Σ to R_Σ.

2.3 Definitions. Given u in R_Σ, $\tilde{u}$ is defined by induction on CH(u):

$$\tilde{u} = \begin{cases} \{\lambda(u)\}, & \delta(u) = 0; \\ \{\lambda(u)\, \tilde{u}_0 \ldots \tilde{u}_{\delta(u)^-}\}, & \delta(u) > 0 \text{ and } SC(u) = \{<>\}; \\ \{\lambda(u_{\alpha_k}) e^k_0 \ldots e^k_{\delta(u_{\alpha_k})^-} : k \in n\}, & \text{otherwise,} \end{cases}$$

where u has root component $\{u_{\alpha_0}, \ldots, u_{\alpha_{n^-}}\}$ and

$$e_j^k = \begin{cases} \tilde{u}_{\alpha_k \cdot j}, \text{ if } \alpha_k \cdot j \text{ represents an acyclic edge of } u \text{ [§1.3];} \\ \tau, \text{ otherwise.} \end{cases}$$

The <u>components</u> of $\tilde{u}$ consist of $\tilde{u}$ (of <u>height</u> $CH(\tilde{u})$) and components of cell elements $e_j^k = \tilde{u}_{\alpha_k \cdot j}$ (of height $CH(\tilde{u}_{\alpha_k \cdot j})$).

<u>Convention</u>. In the following procedure, $\tilde{u}$ is represented as a list U of components represented in order of increasing component height (i.e., $\tilde{u}$ corresponds to the last entry). The k-<u>th</u> component U(k) is a set of one or more <u>cells</u>, each consisting of a label and a list of indices $j \leq k$; the index k in U(k) represents a cyclic edge (τ) in $\tilde{u}$. Similarly for $\tilde{v}$ and V.

2.4 <u>Procedure</u>. SE(u, v) where u, v ε R_Σ, returns ($u \lesssim v$).

```
Let U, V be the indexed arrays representing ũ, ṽ.
Variable A : [|U| x |V| → Bool], initially A(i,j) = False;
[A(i,j) = True iff U(i) appears (has been embedded) in V(j)]
For k ← U to |V|⁻:
    [ [Propogate appearances from proper components of V(k)]:
          For c'w' in V(k):
                For j ← 0 to |w'|⁻:
                      A(*, k) ← A(*, k) v  A(*, w'ⱼ);
          [Scan U for new components embeddable in V(k)]:
          For j ← 0 to |U|⁻:
                [If ~A(j,k), then [test for new component embedding]
                      If Embeds (A, U, j, V, k) then
                            [A(j, k) ← True;
                                  If j = |U|⁻ then Return (True) [ũ embedded];
                ]
    ]
```

13 <u>Return</u> (False) [$\tilde{u}$ never embedded in $\tilde{v}$]

<u>end</u> SE

<u>Remarks</u>

1. The propogation step #6 requires $O(|U|)$ elementary operations for each j, k, value, or $O(\langle u \rangle \times \langle v \rangle)$ operations in all.

2. The embedding test #10 essentially implements $\tilde{\gtrsim}$ on $\tilde{S}_{\Sigma}$, treating acyclic edges leaving U(j) as pointers into "leaves" previously embedded (as recorded by A) in V(k) and its components.

3. Step 1 requires $O(\langle u \rangle + \langle v \rangle)$ operations, given adjacency structures for u and v [36].

2.5 Procedure . Embeds (A, U, j, V, k): Boolean;

```
Variable match: Boolean ← False;
For cw in U(j):
    [For c'w' in V(k) until match:
        If c ≳ c' then
            If |w| > 0 then
                If |w| < |w'| then
                    [For p, q ← 0 while [p < |w| ∧ [p = |w|⁻ ∨ q <
                                |w'|⁻]:
                        [q ← q⁺ until [L(w_p, w'_q) ∨ q = |w'|];
                            If q < |w'| then [p ← p⁺;
                                q ← q⁺]];
                    If p = |w| then match ← True]
                else match ← False
            else match ← True
        else match ← False [end  inner loop];
    If ~ match then Return (False)];
Return (True) [each cell matched];
L(m,n) = [If m = j then [cyclic edge in U(j)]
                            [n = k] [cyclic edge in V(k) required]
            else [ m < j, acyclic edge]
                            A(m,n) [even if n = k, after propogation]]
end Embeds
```

Remarks

1. $|U(j)| \leq$ number of cyclic edges in j-th component +1; similarly for $|V(k)|$.

2. Matching algorithm #4..13 [#7..9] requires $O(|w| + |w'|) \leq O(|w| \times |w'|)$ elementary operations to find η of Definition 1.13 (successive q values at #9 if $p = |w|$ (complete match) at end).

3. Embeds (A, U, j, V, k) is executed $\leq$ once for each $j \varepsilon |U|$, $k \varepsilon |V|$, requiring $O(e_j \times e_k)$ elementary operations each execution ($e_j = \Sigma(|w|$ for cw in U(j)), $e_k = \Sigma(|w'|$ for c'w' in V(k)).

2.6 Theorem. SE(u, v) = True iff $u \lesssim v$, and SE(u, v) returns True or False within $O(\langle u\rangle \times \langle v\rangle)$ elementary operations.

Indication of proof. The time bounds are established by the remarks following Procedures 2.4 and 2.5. Correctness is established by induction on CH(u), in effect replacing U by some prefix U' and showing that SE(u', v) correctly records in A each embedding of each component of U' in a component of V. For the induction base, verify that if U' represents a singleton object c in Σ then Procedure 2.4 (with j restricted to $0 \leq |U|^{-}$) sets A(0, k) = True iff $c \lesssim c'$ where c' occurs as a cell label in some component of V(k). Embeds detects these appearances even if V(k) = {c'} of component height 0. The bottom-to-top order of components in U is critical for this argument: in Steps #8..12 several nested components of U may be embedded in a single component V(k); the lower level embeddings must be recorded in A(*, k) so as to satisfy the L-predicate of Embeds when the higher level embeddings are tested.∎

3. Canonical Simplification

This section applies the wqo preservation result of §1 to the design of canonical (rewrite-rule based) simplifiers for a typed λ-applicative syntax $\Lambda_{\underline{\Sigma}}$. $\underline{\Sigma}$ consists of a vocabulary $\Sigma = C_\Sigma \cup V_\Sigma$ (constants and variables) and an assignment $\tau_\Sigma = \Lambda_\Sigma \rightarrow \tau^*_\Sigma$ of types in a lower semi-lattice $(\tau^*_\Sigma, \subseteq_\Sigma, \perp)$ with minimal element $\perp$ the null ("undefined") type. An abstraction $\lambda x.u$ in Λ_Σ is interpreted (details omitted) as a function defined only on objects of type $\tau_\Sigma(x)$ (a subdomain of the interpretation's structure). Type free λ-calculus results from inclusion of universal-typed variables, and a first-order language results from allowing only individual sorted variables in V_Σ. C_Σ is assumed to include the logical constant "=", and an <u>equation</u> $((=u)v)$ in Λ_Σ is abbreviated $[u = v]$.

A canonical simplifier for Λ_Σ uses a finite set E of equations as term rewriting or replacement rules in addition to the appropriate λ-conversion schemas. Often it is necessary to decide whether the reducibility relation $\geq_E$ is well founded in the sense that $u \geq_E v$ for some $\geq_E$-minimal term $v (u \in \Lambda_\Sigma)$. v is said to be $\geq_E$-minimal if $v \geq_E v'$ implies $v' \geq_E v$. (In practice the set of $\geq_E$-minimal terms should be decidable.)

The principal result of this section is that $\geq_E$ is well founded iff $\geq_E \subseteq \tilde{\trianglerighteq}$ for some quasi-simplification ordering (qso) $\tilde{\trianglelefteq}$. These orderings formalize both syntactical and semantical concepts of complexity. The smallest qso on Λ_Σ is the tree-embedding relation $\tilde{\trianglelefteq}_\Sigma$ on Λ_Σ. It has the property that $\tau_\Sigma(x) \subseteq_\Sigma \tau_\Sigma(y)$ implies $\lambda x.u \tilde{\trianglelefteq}_\Sigma \lambda y.[y/x]u$ (y not free in u). In other words, the "less defined" restriction of two otherwise similar abstractions is considered simpler.

The λ-applicative syntax described below has an unconventional representation by digraphs containing cyclic edges which represent occurrences of bound variables. The terminus of each such edge indicates the type of the bound variable. Moreover, recursion can be represented directly in the "rational closure" $\bar{\Lambda}_\Sigma$ of Λ_Σ by cyclic edges. We assume some familiarity with λ-calculus [13], relying on a few examples and a fuller treatment of Λ_Σ in [4]. Conventions already mentioned are illustrated by the factorial function f:Nat $\to$ Nat defined by f(n) = [if n = 0 then 1 else [n x f(n^-)]], or by the λ-applicative term ($Y\lambda f.\lambda n$:Nat.($\supset$[n = 0] 1 [n x (fn^-)])) where Y is Turing's minimal fixed point operator,

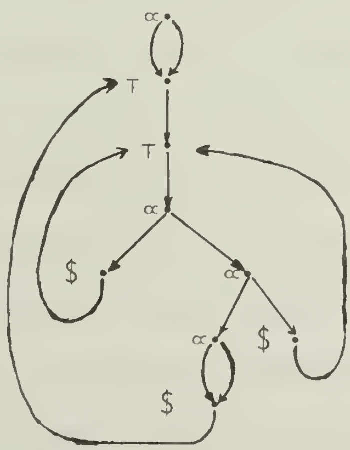
$$Y = [\lambda x.\lambda x.y(xxy)][\lambda x.\lambda y.y(xxy)]$$

characterized by

$$Yu \geq_{\underline{\lambda}_\Sigma} u(Yu) \qquad (u \in \Lambda_\Sigma).$$

where $\underline{\lambda}_\Sigma$ is the set of instances of the λ-conversion laws for a given compactly typed algebraic signature Σ.

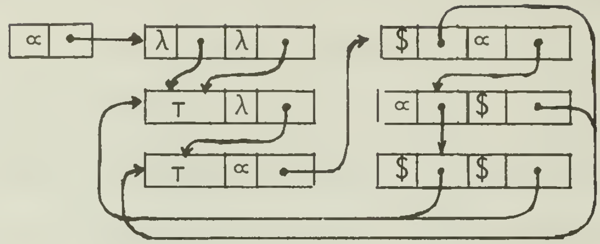
In the λ-applicative syntax Λ_Σ the term Y is



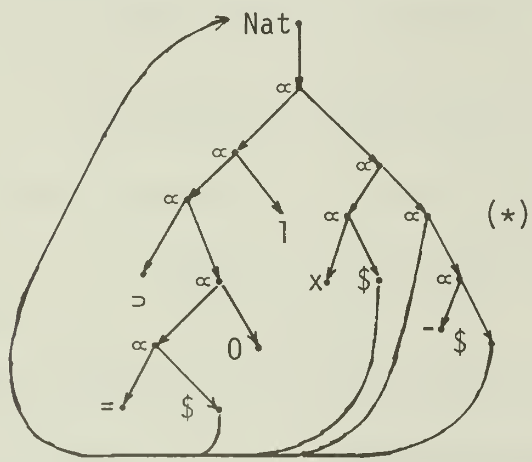
(5)

where the bound variables are of "universal" type τ. A natural computer representation for (5) would use four types of tagged pointers (atom,

bound variable, application, abstraction):



In the rationally closed syntax Λ_Σ^- [8] the factorial function is represented by $(Y\lambda f.\lambda n{:}Nat.\supset[n = 0]\ 1\ [n \times fn^-])^-$:



(6)

where $\supset$ is the "conditional" operation constant. Note the distinction between bound variable references ($) and the recursive invocation (*) of "factorial." In Λ_Σ^- each occurrence of $(Y\lambda f.u)$ in Λ_Σ is replaced by u' wherein each occurrence of f within u is replaced by a cyclic reference to the root of u'. We mention (but do not formalize) Λ_Σ^- because it provides a natural basis for reasoning about recursive program schemas [5], where distinct "unfoldings" of a recursive schema have isomorphic infinite tree representations. The concepts and results stated below for Λ_Σ are easily extended to Λ_Σ^- on the basis of §1.

3.1 Definitions. A compactly typed signature (cts) is a system $\underline{\Sigma}$ = $(\Sigma, \tau_\Sigma, \subseteq_\Sigma, \perp)$ where $\tau_\Sigma : \Lambda_\Sigma \to \tau^*_\Sigma (\tau_\Sigma : \Sigma \to \tau^*_\Sigma$ in particular) and $(\tau^*_\Sigma, \subseteq_\Sigma, \perp)$ is a lower semilattice with minimal element $\perp$, staisfying (a) - (d):

(a) If $\bigcap_\Sigma S = \perp$ then $\bigcap_\Sigma S' = \perp$ for some finite S' $S(S \subseteq \tau^*_\Sigma)$.

(b) If $\tau_\Sigma(u_k) \subseteq_\Sigma \tau_\Sigma(v_k)$ (k = 0, 1) then $\tau_\Sigma(u_0 u_1) \subseteq_\Sigma \tau_\Sigma(v_0 v_1)$.

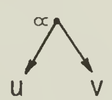
(c) If $\tau_\Sigma(x) \subseteq_\Sigma \tau_\Sigma(y)$ and $\tau_\Sigma(u) \subseteq_\Sigma \tau_\Sigma(v)$ $(x, y \in V_\Sigma;\ u, v \in \Lambda_\Sigma$ where y not free in v) then $\tau_\Sigma(\lambda x.u) \subseteq_\Sigma \tau_\Sigma(\lambda y.[y/x]v)$.

(d) $\Sigma = C_\Sigma \cup V_\Sigma$ (drawn from two disjoint classes of "atomic" symbols), and $\{y : \tau_{\underline{\Sigma}}(y) = \tau_{\underline{\Sigma}}(x)\}$ is infinite for each variable x in V_Σ.

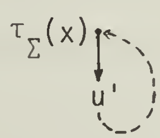
Λ_Σ is the subspace of $R_{(\Sigma \cup \{\propto, \$\} \cup \tau^*_\Sigma)}$ [§1] defined inductively by

(e) $\Sigma \subseteq \Lambda_\Sigma$ (identifying elements of Σ with their singleton graphs);

(f) if $u, v \in \Lambda_\Sigma$ then Λ_Σ contains (u v):



(g) if $x \in V_\Sigma$, $u \in \Lambda_\Sigma$, and V_Σ contains variables of functional type $[\tau_\Sigma(x) \to \tau_\Sigma(u)]$, then Λ_Σ contains $\lambda x.u$:



where $\lambda(\lambda x.u) = \tau_\Sigma(x)$ and u' is the result of replacing each edge terminating at (a vertex labeled by) x by a unary vertex labeled with \$ and outgoing edge terminating at $\rho(\lambda x.u)$. Note that $\lambda(\lambda x.u)$ = label $(\tau_\Sigma(x))$ of $\rho(\lambda x.u)$ as defined in §1; the two uses of λ are unrelated and contextually unambiguous.

<u>Notation</u>. We employ the usual abbreviations: $(t_0 t_1 \ldots t_n)$ for $(\ldots(t_0 t_1)\ldots tn)$, $\lambda x_0 x_1 \ldots x_n . t$ for $\lambda x_0 . (\lambda x_1 \ldots x_n . t)$.

<u>Remarks</u>

1. The definition itself does not specify how τ_Σ is extended from Σ to Λ_Σ. Normally, τ^*_Σ consists of sorts (subtypes of ι_Σ, the class of individuals), functional types $[\alpha \to \beta]$ and subtypes of these, and generic types (such as the universal type $\top$) which properly include functional types. Condition (g) limits λ-terms to types for which there are variables, resulting in a more or less standard first-order applicative syntax with <u>no</u> abstractions when only individual sorted variables are in V_Σ.

2. The lattice meet operation $\cap_\Sigma$ obtained from $\subseteq_\Sigma$ should be effective; otherwise, a first-order (typed) unification algorithm cannot be defined [4].

3. We may (but need not below) assume that the greatest lower bound $\bigcap_\Sigma S$ is defined for all $S \subseteq \tau^*_{\underline{\Sigma}}$. Without condition (a) we would not have a first-order language: let $S = \{\alpha_k : k \in w\}$ and consider the equations $E_S = \{[x_k = x_{k+1}] : k \in w\}$ where $\tau_{\underline{\Sigma}}(x_k) = \alpha_k$. If $\bigcap S = \bot_\Sigma$ then E_S cannot be satisfied by any $\underline{\Sigma}$-structure. If $\bigcap_\Sigma S' \neq \bot_\Sigma$ for every finite $S' \subseteq S$ then the corresponding subsystems $E_{S'}$ are satisfiable, contrary to the first-order compactness principle. First-order adequacy is all that can be required of an <u>effective</u> calculus, even if a standard second-order model [34] provides the intuitively natural denotational semantics [12].

4. Something should be said about types of combinations $((\lambda x.u)t)$ where $\tau_\Sigma(t) \not\subseteq_\Sigma \tau_\Sigma(x)$. One could simply forbid such combinations when $\tau_\Sigma(t) \cap_\Sigma \tau_\Sigma(x) = \bot$, as in finite simple type theory. More satisfactory (in general) is the convention that $\tau_\Sigma((\lambda x.u)t)$ is a type of objects which includes $\tau_\Sigma(u)$, for interpretations wherein t has type $\tau_\Sigma(x)$, and the type of $\bot$ ("undefined"),

for interpretations wherein t does not.

From a cts $\underline{\Sigma}$ we obtain a quasi-ordering $\tilde{\triangleleft}_{\Sigma}$ on $\Lambda_{\underline{\Sigma}}$.

3.2 <u>Definition</u>. Let $\tilde{\triangleleft}_{\Sigma}$ be the identity relation on $\Sigma \cup \{\$, \propto\}$, extended to τ^{*}_{Σ} by $\alpha \tilde{\triangleleft}_{\Sigma} \beta$ iff $\alpha \leq_{\Sigma} \beta$. Extend $\tilde{\triangleleft}_{\Sigma}$ to Λ_{Σ} (or even R_{Σ}, in general) by

$$u \tilde{\triangleleft}_{\Sigma} v \text{ iff there exists an embedding } h:\hat{u} \tilde{\triangleleft}_{\Sigma} \hat{v} \text{ [§1.6].}$$

3.3 <u>Definition</u>. E_{Σ} is the set of "grammatical" substitutions $\theta = [t\ /x_1,\ldots,t_n/x_n]$ where $\tau_{\Sigma}(t_k) \subseteq_{\Sigma} \tau_{\Sigma}(x_k)$ $(k = 1,\ldots,n)$. θ is the endomorphism on Λ_{Σ} wherein $\theta x_k = t_k$ and $\theta x = x (x \in V_{\Sigma} - \{x_1,\ldots, x_n\})$. Thus, $([t/x]\lambda x.u) = \lambda x.u$ because x does not occur in $\lambda x.u$.

Next, we consider quasi-orderings induced by term-replacement operations based on equational systems $E \subseteq \Lambda_{\Sigma}$.

3.4 <u>Definition</u>. Given a term u in Λ_{Σ} and a position α in $D(u)$, $u[t/\alpha]$ is the result of replacing u_{α} by t in u. Specifically, $D(u[t/\alpha]) = (D(u) - \{\alpha.\beta:\beta \in D(u_{\alpha})\}) \cup \{\alpha.\beta:\beta \in D(t)\}$ and

$$(u[t/\alpha])_{\gamma} = \begin{cases} \hat{u}_{\gamma}, & \text{if } \alpha \text{ is not a prefix of } \gamma; \\ \hat{t}_{\beta}, & \text{if } \gamma = \alpha.\beta. \end{cases}$$

Given a set E of equations, $\geq_E$ is the smallest quasi-ordering on Λ_{Σ} (or Λ^{-}_{Σ}) such that $u \geq_E u[\theta t/\alpha]$ for each $[s = t]$ in E, θ in E_{Σ}, α in $D(u)$ such that $u_{\alpha} = \theta s$. $u \simeq_E v$ iff $u \geq_E v \geq_E u$.

<u>Remarks</u>

1. Of particular relevance is the $\underline{\lambda}_{\Sigma}$-conversion relation $\geq_{\underline{\lambda}_{\Sigma}}$ where $\underline{\lambda}_{\Sigma}$ consists of all instances in Λ_{Σ} of the β-conversion and (type-free) η-conversion schemas:

$$\beta_{\underline{\Sigma}} : \quad [(\lambda x.u)t = [t/x]u] \quad (\tau_\Sigma(t) \underline{\subset} \tau_\Sigma(x));$$

$$[(\lambda x.u)t = \bot] \qquad (\tau_\Sigma(t) \cap_\Sigma \tau_\Sigma(x) = \bot);$$

$$\eta_\Sigma : \quad [\lambda x.(ux) = u] \qquad (x \text{ not free in } u, \tau_\Sigma(x) = \bot).$$

The case where $\tau_\Sigma(t) \cap_\Sigma \tau_\Sigma(x) \neq \bot$ and $\tau_\Sigma(t) \not\subseteq_\Sigma \tau_\Sigma(x)$ is analogous to situations which require "run-time type checking" in strongly typed programming languages with (polymorphic) type inclusions [29, 33]. In this event $(\lambda x.u\ t)$ should perhaps reduce to a conditional expression - e.g., $(\supset(\tau_\Sigma(x)t)([t/x]u)\bot)$ where $\tau_\Sigma(x)$ is used as a monadic predicate, $\supset\top\ xy = x$, and $\supset\bot\ xy = y$.[2]

2. Recall that $\geq_E$ is said to be well founded iff $u \geq_E v$ where $v \geq_E v'$ implies $v \geq_E v'$ $(u \in \Lambda_\Sigma)$. Well foundedness (rather than finite termination) is appropriate when E contains such equations as $[x+y = y+x]$.

3.7 Definition. A quasi-simplification ordering (qso) on Λ_Σ is a quasi-ordering $\lesssim$ such that

(i) $u_\alpha \lesssim u(\alpha \in D(u))$;

(ii) if $u \lesssim u'$ and $v \lesssim v'$ then $(uv) \lesssim (u'v')$;

(iii) if $\tau_\Sigma(x) \subseteq_\Sigma \tau_\Sigma(y)$ and $u \lesssim v$ where y does not occur in v, then $\lambda x.u \lesssim \lambda y.[y/x]\ v$, and

(iv) if $u \lesssim v$ then $\theta u \lesssim \theta v$ $(\theta \in E_\Sigma)$.

$\lesssim$ is simplification ordering if, in addition, it totally orders the set of closed (ground, variable-free) terms in $\Lambda_\Sigma(\Lambda_\Sigma^-)$.

Similar orderings have been investigated previously for standard first-order term languages [2, 6, 31]. The condition (iii) preserves semantic inclusions of functions denoted by λ-terms. Normally, these orderings are required to be well founded (at least modulo $\simeq$)--often a tedious

property to verify [16]. As we shall see, this additional assumption is usually superfluous [§§3.9, 3.10].

3.8 Lemma. $\tilde{\trianglelefteq}_\Sigma$ is the smallest qso on Λ_Σ.

Proof. Suppose $\precsim$ a qso. We argue by induction on terms that $\tilde{\trianglelefteq}_\Sigma \subseteq \precsim$ and $\tilde{\trianglelefteq}_\Sigma$ is a qso.

Clearly $\tilde{\trianglelefteq}_\Sigma$ satisfies (i), (ii), (iv); if u $\tilde{\trianglelefteq}_\Sigma$ v by these rules and the fact that $\tilde{\trianglelefteq}_\Sigma$ is a quasi-ordering, then u $\precsim$ v. Suppose $\tau_\Sigma(x) \subseteq_\Sigma \tau_\Sigma(y)$ and u $\tilde{\trianglelefteq}_\Sigma$ v where (by induction) u $\precsim$ v has also been verified. It suffices to verify (iii), that $\lambda x.u \tilde{\trianglelefteq}_\Sigma \lambda y.[y/x]v$. That $\lambda x.u \tilde{\trianglelefteq}_\Sigma \lambda x.v$ easily follows from existence of an embedding h: $\hat{u} \tilde{\trianglelefteq}_\Sigma \hat{v}$: any occurrence of x in u has a corresponding occurrence in v, and both are replaced by cyclic edges from a new vertex labeled by \$ to the new root vertex labeled by $\tau_\Sigma(x)$. That $\lambda x.v \tilde{\trianglelefteq}_\Sigma \lambda y.[y/x]v$ is also immediate from the assumptions; the conclusion follows by transitivity of $\tilde{\trianglelefteq}_\Sigma$.∎

3.9 Corollary. Suppose $T \subseteq \Lambda_\Sigma(\Lambda_\Sigma^-)$ such that $\{\lambda(u_\alpha):\alpha \in D(u)\ \&\ u \in T\}$ is a finite subset of $\Sigma \cup \tau_\Sigma^*$. Then $(T, \precsim)$ is wqo for any qso $\precsim$.

Proof. $(T, \precsim)$ is a homomorphic image of $(T, \tilde{\trianglelefteq}_\Sigma)$ by Lemma 3.8. Thus by Lemma 1.8(d) it suffices to show that $(T, \tilde{\trianglelefteq}_\Sigma)$ is wqo. Now T is a set of finite digraphs over the finite vocabulary $\Sigma'' = \{\lambda(u_\alpha):\alpha \in D(u)\ \&\ u \in T\}$, which is wqo by $\tilde{\trianglelefteq}_\Sigma$ if we set $\beta \tilde{\trianglelefteq}_\Sigma \gamma$ for all $\beta \subseteq_\Sigma \gamma \in \tau_\Sigma^*$. Thus $(R_{\Sigma''}, \tilde{\trianglelefteq}_\Sigma)$ is wqo by Theorem 1.7, and $(T, \tilde{\trianglelefteq}_\Sigma)$ is wqo because $T \subseteq R_{\Sigma''}$.∎

Given a qso $\tilde{\trianglelefteq}$ on Λ_Σ let $\trianglerighteq_E = \tilde{\trianglerighteq} \cap \geq_E$. $\trianglerighteq_E$ is generated by the set of instances $[\theta s = \theta t]$ of equations in E such that $\theta s \tilde{\trianglerighteq} \theta t$. Define u $\simeq_E$ v iff u $\trianglerighteq_E$ v $\trianglerighteq_E$ u.

3.10 Theorem.[3] Suppose $(E - \lambda_\Sigma)$ finite and $[s = t] \in E$ implies that each free variable of t occurs in s. Then $\geq_E$ is well founded iff there exists a qso

$\tilde{\trianglelefteq}$ such that $\underset{E}{>} = \underset{E}{\rhd}$.

Proof. The theorem holds for both Λ_Σ and Λ_Σ^- ; in the latter case we assume λ_Σ contains the reductions $[Y\lambda x.u = (Y\lambda x.u)^-]$. Note that if $u \underset{\lambda_\Sigma}{>} v$ then each label appearing in v also appears in u. Also, the set of all equations $[(\lambda x.s)t = [t/x]s]$ in λ_Σ whose left-hand side occurs in u is finite. Thus the set of instances $[\theta s = \theta t]$ of equations in E where θs occurs in u is finite, and by the assumption on variables of t no new variables are introduced.

Now suppose $\underset{E}{>} = \underset{E}{\rhd}$ where $\tilde{\trianglelefteq}$ is a qso. It suffices to show that $(T_u, \tilde{\triangleleft})$ is wqo where $T_u = \{v : u \underset{E}{\rhd} v\}$ $(u \in \Lambda_\Sigma^-)$. T_u satisfies the condition of Corollary 3.9 by the preceding remarks. Thus $(\Lambda_\Sigma \ \underset{E}{\lhd})$ is well founded because each $(T_u, \tilde{\trianglelefteq})$ is well founded.

Conversely, suppose $\underset{E}{>}$ is well founded. Let $\tilde{\trianglelefteq}$ be the reflexive and transitive closure of $(\tilde{\triangleleft}_\Sigma \cup \underset{E}{<})$. It is not difficult to verify on the basis of Lemma 3.8 that $\tilde{\trianglelefteq}$ is a qso, and it is evident that $\underset{E}{>} = \underset{E}{\rhd}$.∎

Remarks

1. The condition on variables in equations of E is typically satisfied. Here, as in [15], it can be removed under certain circumstances. Also, if C_Σ is finite and $(\tau_\Sigma^*, \subseteq_\Sigma, \perp)$ is a wqo set then the finiteness conditions on $E - \lambda_\Sigma$ can be removed.

2. Given a qso $\tilde{\trianglelefteq}$, equations in E may be reordered so as to obtain an equivalent system E' such that $t \not\tilde{\rhd} s$ for each equation $[s = t]$ in E'. Effective methods have been developed for transforming E into an equivalent system E' such that $\underset{E'}{\rhd}$ is confluent (on ground terms u_k):

$$u_0 =_E u_1 \text{ implies } u_k \trianglerighteq_{E'} v \ (k = 0, 1) \text{ for some } v.$$

These methods are useful in mechanized reasoning with equality [3, 19, 20].

4. Conclusions

It has been shown that finite ordered digraphs over a wqo vertex label alphabet are wqo by the homeomorphic embedding relation on their infinite tree representations. This relation can be computed in quadratic time; it is the smallest of a useful class of (quasi) simplification orderings used by term replacement systems.

These results provide a practical basis for comparing structural complexity of program schemas and other naturally cyclic finite objects. Theorem 3.10 might be extended in several directions. For example, when is the set of $\geq_E$-minimal terms decidable, and when is $\simeq_E$ decidable on this set? What can be said about effectiveness of a qso $\tilde{\trianglelefteq}$ for which $\geq_E = \trianglerighteq_E$?

Acknowledgement. A connection between simplification orderings and finite tree embedding was brought to the author's attention by D. Plaisted and N. Dershowitz. The latter contributed numerous helpful discussions and criticisms culminating in the argument of §1.

NOTES

[1]Personal communication.

[2]These conventions are based on a two-valued truth-value lattice wherein   represents "undefined", "null type", and "false", and represents both "universal type" and "true". This lattice is related to the more usual three and four-valued truth-value domains [34] in [4].

[3]This argument generalizes the argument for finite acyclic objects due to N. Dershowitz [7]. Note that if $\trianglerighteq_E = \geq_E$ and there exists no infinite descending chain $(u_k : k \in w)$ where $u_k \triangleright u_{k+1}$ $(k \in w)$ then $\trianglerighteq_E$ has the finite termination property (ftp). Conversely, if $\geq_E$ satisfies ftp, then $\geq_E = \trianglerighteq_E$ where $\tilde{\triangleleft}$ is a qso which admits no infinite descending chain.

1. Böhm, C. (ed.), λ-Calculus and Computer Science Theory, Lecture Notes in Computer Science 37 (edited by G. Goos and J. Hartmanis); Springer-Verlag, New York, 1975.

2. Brown, T. C., A Structured Design Method for Specialized Proof Procedures (Ph.D. Thesis), California Institute of Technology, 1974 (University Microfilms, Anne Arbor, Michigan).

3. Brown, T. "Equational reasoning with derived retractions." Research Note, University of Illinois at Urbana-Champaign, April 1979.

4. Brown, T., "Reduction systems with compactly typed operator signatures," in preparation.

5. Burstall, R. M., and Darlington, J. "A transformation system for developing recursive programs," JACM 24 (1977), 44-67

6. Dershowitz, N., and Manna, Zohar, Proving Termination with Multiset Orderings, Standford AI Lab., Memo AIM-310 (March 1978).

7. Dershowitz, N., A note on simplification orderings. UIUC, March 1979.

8. Ehrich, H. D. "Outline of an algebraic theory of structured objects. Automata, Languages, and Programming (edited by S. Michelson and R. Milner, Edinburgh University Press (1976), 508-530.

9. Ehrig, H., H. J. Kreowiski, A. Maggiolo-Schettini, B. K. Rosen, and J. Winkowski, "Deriving structures from structures," Math. Found. Comp. Sci. 1978 (J. Winkowski, ed.), Lecture Notes in Comp. Sci. 64, Springer-Verlag.

10. Garey, M. R., and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, San Francisco, 1979.

11. Goguen, J. A., J. W. Thatcher, E. G. Wagner, and J. B. Wright, "Initial algebra semantics and continuous algebras," JACM 24 (1977), 68-95.

12. Harel,D. Arithmetical completeness in logics of programs, in _5th Automata, Languages, and Programming_ Symposium, Springer-Verlag, 1978.

13. Hindley, R., B. Lercher, and J. P. Seldin, _Introduction to Combinatory Logic_ (Cambridge: University Press, 1972).

14. Hoffman, C. M., and M. J. O'Donnell, "An interpreter generator using tree pattern matching," _6th Annual ACM Symposium on Principles of Programming Languages_ (San Antonio, Texas, 1979), pp. 169-179.

15. Huet, G., "Confluent reductions: abstract properties and applications to term rewriting systems," _Proc. 18th Annual IEEE Symposium on Foundations of Comp. Sci_.

16. Knuth, D., and P. Bendix, "Simple word problems in universal algebras," _Computational Problems in Abstract Algebra_, J. Leech, Ed., Pergammon Press, 1970.

17. Kruskal, J.,"Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture." _Trans. Amer. Math Soc. 95_ (1960), 210-225.

18. Kruskal, J., "The theory of well-quasi-ordering: A frequently discovered concept," _J. Combinatorial Theory Ser. A 13_ (1972), 297-305.

19. Lankford, D., "Canonical inference," Report ATP-32, Math Dept., University of Texas, Austin, 1975.

20. Lankford, D., and Ballentyne, A. M. Decision Procedures for Simple Equational Theories with Commutative Axioms,

21. Laver, R., "On Fraisse's order type conjecture," _Ann. of Math. 93_ (1971), 89-111.

22. Laver, R., "An order type decomposition theorem," _Ann. of Math. 98_ (1973), 96-119.

23. Laver, R., "Better quasi-orderings and a class of trees," Studies in Foundations and Combinatories Advances in Mathematics Supplementary Studies, Vol. 1 (reprint).

24. Levy, M. R., Data Types with Sharing and Circularity (Ph.D. Thesis), Technical Report CS-78-26, Dept. Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1978.

25. Levy, M. R., and T. S. E. Maibaum, "Continuous data types," Research Report CS-79-08, Dept. Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1979.

26. Nash-Williams, C., St. J. A., "On Well-quasi-ordering finite trees," Proc. Cambridge Philos. Soc. 59 (1963), pp. 833-835.

27. Nash-Williams, C. St. J. A., "On well-quasi-ordering infinite trees," Proc. Camb. Phil. Soc. 61 (1965), 697-720.

28. Nash-Williams, C. St. J. A., "On better quasi-ordering transfinite sequences," Proc. Camb. Phil. Soc. 64 (1968), 273-290.

29. O'Donnell,M.J., "A programming language theorem which is independent of Peano Arithmetic," Proc. 11th Annual ACM Symposium on Theory of Computing,1979

30. Ore, Oystein, Theory of Graphs, American Mathematical Society Colloquium Publications, Vol. 38 (1962).

31. Plaisted, D., Well-founded Orderings for Proving Termination of Systems of Rewrite Rules, Report UIUCDCS-R-78-932, Dept. Comp. Science, University of Illinois at Urbana-Champaign (July 1978).

32. Rado, R., Partial well-orderings of sets of vectors, Mathematika 1 (1954), 89-95.

33. Reynolds, J. C. "Towards a theory of type structure," Colloquium on Programming, Paris, 1974.

34. Scott, D.,"Data types as lattices," SIAM J. Comput. 5 (1976), 522-585.

35. Shonfield, J., Degrees of Unsolvability. North Holland, Amsterdam, 1971.

36. Tarjan, R. E., "Depth first search and linear graph algorithms," SIAM J. Comput. 1 (1972), 146-160.

37. Wagner, E. G., J. W. Thatcher, J. B. Wright, "Programming languages as mathematical objects," Mathematical Foundations of Computer Science 1978 (J. Winkowski, Ed.), Lect. Notes in Comput. Science 64, pp. 84-101.

38. Wegbreit, Ben, "Proving properties of complex data structures," JACM 23 (1976) pp. 389-396.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-79-981	2.	3. Recipient's Accession No.
4. Title and Subtitle Canonical Simplification of Finite Objects Well Quasi-Ordered by Tree Embedding			5. Report Date August 1979
			6.
7. Author(s) Thomas C. Brown, Jr.			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Dept. of Computer Science University of Illinois Urbana, Illinois 61801			10. Project/Task/Work Unit No.
			11. Contract/Grant No. NSF-MCS-77-22830
12. Sponsoring Organization Name and Address National Science Foundation Washington, D.C.			13. Type of Report & Period Covered
			14.

15. Supplementary Notes

16. Abstracts

A finite object space can be viewed as the set of finite and infinite (edge-) ordered trees representable as finite ordered vertex-labeled digraphs. We show that the order-preserving homeomorphic tree-embedding relation on finite objects over a well quasi-ordered (wqo) label alphabet is wqo. Canonical simplifiers require well-founded object-orderings to orient and ensure finite termination of replacement rules. We characterize these orderings as homomorphic refinements of the wqo tree-embedding relation, and we show this relation's time complexity to be O(mxn) where m and n are the edge counts of the digraph representations.

17. Key Words and Document Analysis. 17a. Descriptors

Well quasi-ordering, simplification ordering, rational (regular) trees, canonical simplification, homeomorphic tree-embedding.

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (10-70) USCOMM-DC 40329-P71